
Online Square Packing

Sándor P. Fekete · Tom Kamphans · Nils
Schweer

Abstract We analyze the problem of packing squares in an online fashion: Given a semi-infinite strip of width 1 and an unknown sequence of squares of side length in $[0, 1]$ that arrive from above, one at a time. The objective is to pack these items as they arrive, minimizing the resulting height. Just like in the classical game of Tetris, each square must be moved along a collision-free path to its final destination. In addition, we account for gravity in both motion (squares must never move up) and position (any final destination must be supported from below). A similar problem has been considered before; the best previous result is by Azar and Epstein, who gave a 4-competitive algorithm in a setting without gravity (i.e., with the possibility of letting squares “hang in the air”) based on ideas of shelf-packing: Squares are assigned to different horizontal levels, allowing an analysis that is reminiscent of some bin-packing arguments. We apply a geometric analysis to establish a competitive factor of 3.5 for the bottom-left heuristic and present a $\frac{34}{13} \approx 2.6154$ -competitive algorithm.

Keywords Online packing · strip packing · squares · gravity · Tetris.

1 Introduction

1.1 Packing Problems

Packing problems arise in many different situations, either concrete (where actual physical objects have to be packed), or abstract (where the space is virtual, e.g., in scheduling). Even in a one-dimensional setting, computing an optimal set of positions in a container for a known set of objects is a classical, hard problem. Having to deal with two-dimensional objects adds a variety of difficulties; one of them is the more complex

Tom Kamphans was supported by DFG grant FE 407/8-3, project “ReCoNodes”. A preliminary extended abstract summarizing the results of this paper appeared in [11].

Braunschweig University of Technology
Department of Computer Science, Algorithms Group
Mühlenpfordtstrasse 23, 38106 Braunschweig, Germany
<http://www.ibr.cs.tu-bs.de/alg>
E-mail: {s.fekete, t.kamphans, n.schweer}@tu-bs.de.

structure of feasible placements; see, for example, Fekete et al. [13]. Another one is actually moving the objects into their final locations without causing collisions or overlap along the way. A different kind of difficulty may arise from a lack of information: in many settings, objects have to be assigned to their final locations one by one, without knowing future items. Obviously, this makes the challenge even harder.

In this paper, we consider online packing of squares into a vertical strip of unit width. Squares arrive from above in an online fashion, one at a time, and have to be moved to their final positions. On this path, a square may move only through unoccupied space, come to a stop only if it is supported from below; in allusion to the well-known computer game, this is called the *Tetris constraint*. In addition, an item is not allowed to move upwards and has to be supported from below when reaching its final position; these conditions are called *gravity constraints*. The objective is to minimize the total height of the occupied part of the strip.

1.2 Problem Statement

Let S be a semi-infinite strip of width 1 and $\mathcal{A} = (A_1, \dots, A_n)$ a sequence of squares with side length $a_i \leq 1$, $i = 1, \dots, n$. The sequence is unknown in advance. A strategy gets the squares one by one and must place a square before it gets the next. Initially, a square is located above all previously placed ones.

Our goal is to find a non-overlapping packing of squares in the strip that keeps the height of the occupied area as low as possible. More precisely, we want to minimize the distance between the bottom side of S and the highest point that is occupied by a square. The sides of the squares in the packing must be parallel to the sides of the strip. Moreover, a packing must fulfill two additional constraints:

Tetris constraint: At the time a square is placed, there is a collision-free path from the initial position of a square (top of the strip) to the square's final position.

Gravity constraint: A square must be packed on top of another square (*i.e.*, the intersection of the upper square's bottom side and the lower square's top side must be a line segment) or on the bottom of the strip; in addition, no square may ever move up on the path to its final position.

1.3 Related Work

In general, a packing problem is defined by a set of items that have to be packed into a container (a set of containers) such that some objective function, *e.g.*, the area where no item is placed or the number of used containers, is minimized. A huge amount of work has been done on different kinds of packing problems. A survey on approximation algorithms for packing problems can be found in [22].

A special kind of packing problem is the *strip packing problem*. It asks for a non-overlapping placement of a set of rectangles in a semi-infinite strip such that the height of the occupied area is minimized. The bottom side of a rectangle has to be parallel to the bottom side of the strip. Over the years, many different variations of the strip packing problem have been proposed: online, offline, with or without rotation, and so on. Typical measures for the evaluation of approximation and online algorithms are the absolute performance and the asymptotic performance ratio.

If we restrict all rectangles to be of the same height, the strip packing problem without rotation is equivalent to the *bin packing problem*: Given a set of one-dimensional items each having a size between zero and one, the task is to pack these items into a minimum number of unit size bins. Hence, all negative results for the bin packing problem, *e.g.*, NP-hardness and lower bounds on the competitive ratio also hold for the strip packing problem; see [15] for a survey on (online) bin packing.

If we restrict all rectangles to be of the same width then the strip packing problem without rotation is equivalent to the *list scheduling problem*: Given a set of jobs with different processing times, the task is to schedule these jobs on a set of identical machines such that the makespan is minimized. This problem was first studied by Graham [16]. There are many different kinds of scheduling problems, *e.g.*, the machines can be identical or not, *preemption* might be allowed or not, and there might be other restrictions such as *precedence constraints* or *release times*; see [7] for a textbook on scheduling.

Offline Strip Packing Concerning the absolute approximation factor, Baker *et al.* [3] introduce the class of *bottom-up left-justified* algorithms. A specification that sorts the items in advance is a 3-approximation for a sequence of rectangles and a 2-approximation for a sequence of squares. Sleator [21] presents an algorithm with approximation factor 2.5, Schiermeyer [19] and Steinberg [23] present algorithms that achieve an absolute approximation factor of 2, for a sequence of rectangles.

Concerning the asymptotic approximation factor, the algorithms presented by Coffman *et al.* [9] achieve performance bounds of 2, 1.7, and 1.5. Baker *et al.* [2] improve this factor to 1.25. Kenyon and Rémila [18] design a fully polynomial time approximation scheme. Han *et al.* [17] show that every algorithm for the bin packing problem implies an algorithm for the strip packing problem with the same approximation factor. Thus, in the offline case, not only the negative results but also the positive results from bin packing hold for strip packing.

Online Strip Packing Concerning the absolute competitive ratio Baker *et al.* [4] present two algorithms with competitive ratio 7.46 and 6.99. If the input sequence consists only of squares the competitive ratio reduces to 5.83 for both algorithms. These algorithms are the first *shelf algorithms*: A shelf algorithm classifies the rectangles according to their height, *i.e.*, a rectangle is in a class s if its height is in the interval $(\alpha^{s-1}, \alpha^s]$, for a parameter $\alpha \in (0, 1)$. Each class is packed in a separate *shelf*, *i.e.*, into a rectangular area of width one and height α^s , inside the strip. A bin packing algorithm is used as a subroutine to pack the items. Ye *et al.* [25] present an algorithm with absolute competitive factor 6.6623. Lower bounds for the absolute performance ratio are 2 for sequences of rectangles and 1.75 for sequences of squares [6].

Concerning the asymptotic competitive ratio, the algorithms in [4] achieve a competitive ratio of 2 and 1.7. Csirik and Woeginger [10] show a lower bound of 1.69103 for any shelf algorithm and introduce a shelf algorithm whose competitive ratio comes arbitrarily close to this value. Han *et al.* [17] show that for the so called *Super Harmonic* algorithms, for the bin packing problem, the competitive ratio can be transferred to the strip packing problem. The current best algorithm for bin packing is 1.58889-competitive [20]. Thus, there is an algorithm with the same ratio for the strip packing problem. A lower bound, due to van Vliet [24], for the asymptotic competitive ratio, is 1.5401. This bound also holds for sequences consisting only of squares.

Tetris Every reader is certainly familiar with the classical game of Tetris: Given a strip of fixed width, find an online placement for a sequence of objects falling down from above such that space is utilized as good as possible. In comparison to the strip packing problem, there is a slight difference in the objective function as Tetris aims at filling rows. In actual optimization scenarios this is less interesting as it is not critical whether a row is used to precisely 100%—in particular, as full rows do not magically disappear in real life. In this process, no item can ever move upward, no collisions between objects must occur, an item will come to a stop if and only if it is supported from below, and each placement has to be fixed before the next item arrives. Even when disregarding the difficulty of ever-increasing speed, Tetris is notoriously difficult: Breukelaar *et al.* [5] show that Tetris is PSPACE-hard, even for the, original, limited set of different objects.

Strip Packing with Tetris Constraint Tetris-like online packing has been considered before. Most notably, Azar and Epstein [1] consider online packing of rectangles into a strip; just like in Tetris, they consider the situation with or without rotation of objects. For the case without rotation, they show that no constant competitive ratio is possible, unless there is a fixed-size lower bound of ε on the side length of the objects, in which case there is an upper bound of $O(\log \frac{1}{\varepsilon})$ on the competitive ratio.

For the case in which rotation is possible, they present a 4-competitive strategy based on shelf-packing methods: Each rectangle is rotated such that its narrow side is the bottom side. The algorithm tries to maintain a corridor at the right side of the strip to move the rectangles to their shelves. If a shelf is full or the path to it is blocked, by a large item, a new shelf is opened. Until now, this is also the best deterministic upper bound for squares. Note that in this strategy gravity is not taken into account as items are allowed to be placed at appropriate levels.

Coffman *et al.* [8] consider probabilistic aspects of online rectangle packing without rotation and with Tetris constraint. If n rectangle side lengths are chosen uniformly at random from the interval $[0, 1]$, they show that there is a lower bound of $(0.31382733\dots)n$ on the expected height for any algorithm. Moreover, they propose an algorithm that achieves an asymptotic expected height of $(0.36976421\dots)n$.

Strip Packing with Tetris and Gravity Constraint There is one negative result for the setting with Tetris and gravity constraint when rotation is not allowed in [1]: If all rectangles have a width of at least $\varepsilon > 0$ or of at most $1 - \varepsilon$, then the competitive factor of any algorithms is $\Omega(\frac{1}{\varepsilon})$.

1.4 Our Results

We analyze a natural and simple heuristic called *BottomLeft* (Section 2), which works similar to the one introduced by Baker *et al.* [3]. We show that it is possible to give a better competitive ratio than the ratio 4 achieved by Azar and Epstein, even in the presence of gravity. We obtain an asymptotic competitive ratio of 3.5 for *BottomLeft*. Furthermore, we introduce the strategy *SlotAlgorithm* (Section 3), which improves the upper bound to $\frac{34}{13} = 2.6154\dots$, asymptotically.

2 The Strategy *BottomLeft*

In this section, we analyze the packing generated by the strategy *BottomLeft*, which works as follows: We place the current square as close as possible to the bottom of the strip; this means that we move the square along a collision-free path from the top of the strip to the desired position, without ever moving the square in positive y -direction. We break ties by choosing the leftmost among all possible bottommost positions.

A packing may leave areas of the strip empty. We call a maximal connected component (of finite size) of the strip's empty area a *hole*, denoted by H_h , $h \in \mathbb{N}$ denote by $|H_h|$ the size of H_h . For a simplified analysis, we finish the packing with an additional square, A_{n+1} , of side length 1. As a result, all holes have a closed boundary. Let H_1, \dots, H_s be the holes in the packing. We can express the height of the packing produced by *BottomLeft* as follows:

$$BL = \sum_{i=1}^n a_i^2 + \sum_{h=1}^s |H_h|.$$

In the following sections, we prove that

$$\sum_{h=1}^s |H_h| \leq 2.5 \cdot \sum_{i=1}^{n+1} a_i^2.$$

Because any strategy produces at least a height of $\sum_{i=1}^n a_i^2$, and because $a_{n+1}^2 = 1$, we get

$$BL = \sum_{i=1}^n a_i^2 + \sum_{h=1}^s |H_h| \leq \sum_{i=1}^n a_i^2 + 2.5 \cdot \sum_{i=1}^{n+1} a_i^2 \leq 3.5 \cdot OPT + 2.5,$$

where OPT denotes the height of an optimal packing. This proves:

Theorem 1 *BottomLeft* is (asymptotically) 3.5-competitive.

Definitions Before we start with the analysis, we need some definitions: We denote the bottom (left, right) side of the strip by B_S (R_S , L_S ; respectively), and the sides of a square, A_i , by B_{A_i} , T_{A_i} , R_{A_i} , L_{A_i} (bottom, top, right, left; respectively); see Fig. 1. The x -coordinates of the left and right side of A_i in a packing are l_{A_i} and r_{A_i} ; the y -coordinates of the top and bottom side t_{A_i} and b_{A_i} , respectively. Let the *left neighborhood*, $N_L(A_i)$, be the set of squares that touch the left side of A_i . In the same way we define the bottom, top, and right neighborhoods, denoted by $N_B(A_i)$, $N_T(A_i)$, and $N_R(A_i)$, respectively.

A point, P , is called *unsupported*, if there is a vertical line segment pointing from P to the bottom of S whose interior lies completely inside a hole. Otherwise, P is *supported*. A section of a line segment is supported, if every point in this section is supported.

For an object ξ , we refer to the boundary as $\partial\xi$, to the interior as ξ° , and to its area by $|\xi|$. If ξ is a line segment, then $|\xi|$ denotes its length.

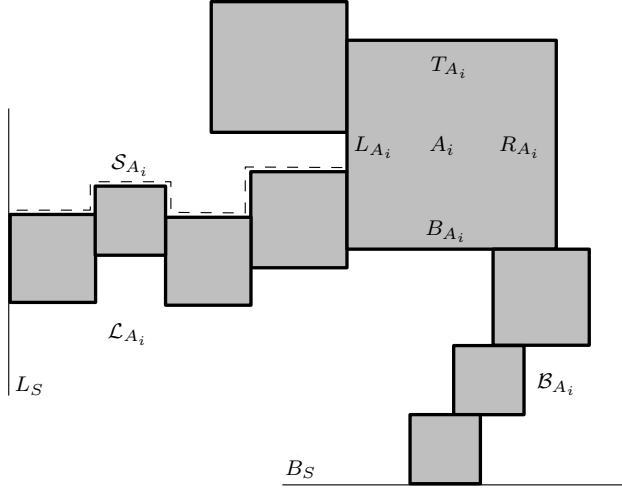


Fig. 1 The square A_i with its left sequence \mathcal{L}_{A_i} , the bottom sequence \mathcal{B}_{A_i} , and the skyline \mathcal{S}_{A_i} . The left sequence ends at the left side of S , and the bottom sequence at the bottom side of S .

Outline of the Analysis We proceed as follows: First, we state some basic properties of the generated packing (Section 2.1). In Section 2.2 we simplify the shape of the holes by partitioning a hole, produced by BottomLeft, into several disjoint new holes. In the packing, these new holes are open at their top side, so we introduce *virtual lids* that close these holes. Afterwards, we estimate the area of a hole in terms of the squares that enclose the hole (Section 2.3). First, we bound the area of holes that have no virtual lid and whose boundary does not intersect the boundary of the strip. Then, we analyze holes with a virtual lid; as it turns out, these are “cheaper” than holes with non-virtual lids. Finally, we show that holes that touch the strip’s boundary are just a special case. Section 2.4 summarizes the costs that are charged to a square.

2.1 Basic Properties of the Generated Packing

In this section, we show some basic properties of a packing generated by BottomLeft. In particular, we analyze structural properties of the boundary of a hole.

We say that a square, A_i , *contributes* to the boundary of a hole, H_h , iff ∂A_i and ∂H_h intersect in more than one point, *i.e.*, $|\partial A_i \cap \partial H_h| > 0$. For convenience, we denote the squares on the boundary of a hole by $\tilde{A}_1, \dots, \tilde{A}_k$ in counterclockwise order starting with the upper left square; see Fig. 2. It is always clear from the context which hole defines this sequence of squares. Thus, we chose not to introduce an additional superscript referring to the hole. We define $\tilde{A}_{k+1} = \tilde{A}_1$, $\tilde{A}_{k+2} = \tilde{A}_2$, and so on. By $P_{i,i+1}$ we denote the point where ∂H_h leaves the boundary of \tilde{A}_i and enters the boundary of \tilde{A}_{i+1} ; see Fig. 3.

Let A_i be a square packed by BottomLeft. Then A_i can be moved neither to the left nor down. This implies that either $N_L(A_i) \neq \emptyset$ ($N_B(A_i) \neq \emptyset$) or that L_{A_i} (B_{A_i}) coincides with L_S (B_S). Therefore, the following two sequences \mathcal{L}_{A_i} and \mathcal{B}_{A_i} exist: The first element of \mathcal{L}_{A_i} (\mathcal{B}_{A_i}) is A_i . The next element is chosen as an arbitrary left

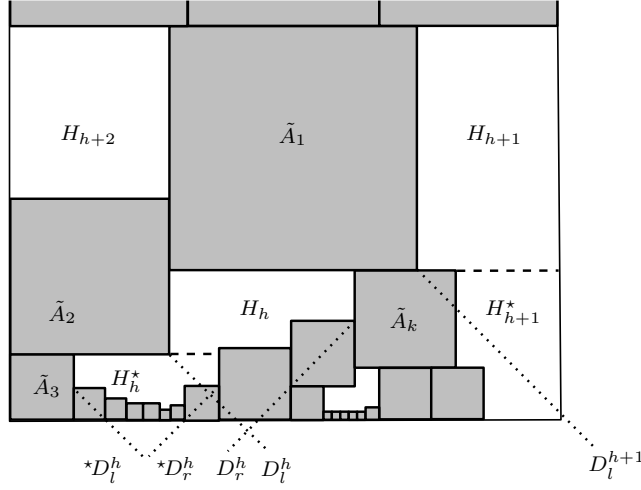


Fig. 2 A packing produced by BottomLeft. The squares $\tilde{A}_1, \dots, \tilde{A}_k$ contribute to the boundary of the hole H_h . In the analysis, H_h is split into a number of subholes. In the shown example one new subhole H_h^* is created. Note that the square \tilde{A}_1 also contributes to the holes H_{h+1} and H_{h+2} . Moreover, it serves as a virtual lid for H_{h+1}^* .

(bottom) neighbor of the previous element. The sequence ends if no such neighbor exists. We call \mathcal{L}_{A_i} the *left sequence* and \mathcal{B}_{A_i} the *bottom sequence* of a square A_i ; see Fig. 1

We call the polygonal chain from the upper right corner of the first element of \mathcal{L}_{A_i} to the upper left corner of the last element, while traversing the boundary of the sequence in counterclockwise order, the *skyline*, \mathcal{S}_{A_i} , of A_i .

Obviously, \mathcal{S}_{A_i} has an endpoint on L_S and $\mathcal{S}_{A_i}^\circ \cap H_h^\circ = \emptyset$. With the help of \mathcal{L}_{A_i} and \mathcal{B}_{A_i} we can prove (see Fig. 3):

Lemma 1 *Let \tilde{A}_i be a square that contributes to ∂H_h . Then,*

- (i) $\partial H_h \cap \partial \tilde{A}_i$ is a single curve, and
- (ii) if ∂H_h is traversed in counterclockwise (clockwise) order, $\partial H_h \cap \partial \tilde{A}_i$ is traversed in clockwise (counterclockwise) order w.r.t. $\partial \tilde{A}_i$.

Proof For the first part, suppose for a contradiction that $\partial H_h \cap \partial \tilde{A}_i$ consists of at least two curves, c_1 and c_2 . Consider a simple curve, C , that lies completely inside H_h and has one endpoint in c_1 and the other one in c_2 . We add the straight line between the endpoints to C and obtain a simple closed curve C' . As c_1 and c_2 are not connected, there is a square, \tilde{A}_j , inside C' that is a neighbor of \tilde{A}_i . If \tilde{A}_j is a left, right or bottom neighbor of \tilde{A}_i this contradicts the existence of $\mathcal{B}_{\tilde{A}_j}$ and if it is a top neighbor this contradicts the existence of $\mathcal{L}_{\tilde{A}_j}$. Hence, $\partial H_h \cap \partial \tilde{A}_i$ is a single curve.

For the second part, imagine that we walk along ∂H_h in counterclockwise order. Then, the interior of H_h lies on our left-hand side, and all squares that contribute to ∂H_h lie on our right-hand side. Hence, their boundaries are traversed in clockwise order w.r.t. their interior.

We define P and Q to be the left and right endpoint, respectively, of the line segment $\partial \tilde{A}_1 \cap \partial H_h$. Two squares \tilde{A}_i and \tilde{A}_{i+1} can basically be arranged in four ways,

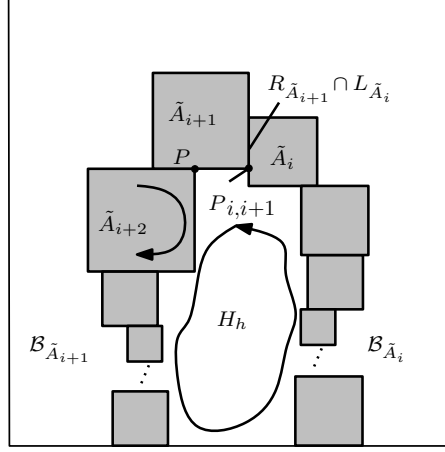


Fig. 3 The hole H_h with the two squares \tilde{A}_i and \tilde{A}_{i+1} and their bottom sequences. In this situation, \tilde{A}_{i+1} is \tilde{A}_1 . If ∂H_h is traversed in counterclockwise order then $\partial H_h \cap \partial \tilde{A}_{i+2}$ is traversed in clockwise order w.r.t. to $\partial \tilde{A}_{i+2}$.

i.e., \tilde{A}_{i+1} can be a left, right, bottom or top neighbor of \tilde{A}_i . The next lemma restricts these possibilities:

Lemma 2 Let $\tilde{A}_i, \tilde{A}_{i+1}$ be a pair of squares that contribute to the boundary of a hole H_h .

- (i) If $\tilde{A}_{i+1} \in N_L(\tilde{A}_i)$, then either $\tilde{A}_{i+1} = \tilde{A}_1$ or $\tilde{A}_i = \tilde{A}_1$.
- (ii) If $\tilde{A}_{i+1} \in N_T(\tilde{A}_i)$, then $\tilde{A}_{i+1} = \tilde{A}_1$ or $\tilde{A}_{i+2} = \tilde{A}_1$.

Proof (i) Let $\tilde{A}_{i+1} \in N_L(\tilde{A}_i)$. Consider the endpoints of the vertical line $R_{\tilde{A}_{i+1}} \cap L_{\tilde{A}_i}$; see Fig. 3. We traverse ∂H_h in counterclockwise order starting in P . By Lemma 1, we traverse $\partial \tilde{A}_i$ in clockwise order, and therefore, $P_{i,i+1}$ is the lower endpoint of $R_{\tilde{A}_{i+1}} \cap L_{\tilde{A}_i}$. Now, $\mathcal{B}_{\tilde{A}_i}$, $\mathcal{B}_{\tilde{A}_{i+1}}$, and the segment of \mathcal{B}_S completely enclose an area that completely contains the hole, H_h . If the sequences have a square in common, we consider the area enclosed up to the first intersection. Therefore, if $b_{\tilde{A}_{i+1}} \geq b_{\tilde{A}_i}$ then $\tilde{A}_{i+1} = \tilde{A}_1$ else $\tilde{A}_i = \tilde{A}_1$ by the definition of \overline{PQ} .

The proof of (ii) follows almost directly from the first part. Let $\tilde{A}_{i+1} \in N_T(\tilde{A}_i)$. If ∂H_h is traversed in counterclockwise order, we know that $\partial \tilde{A}_{i+1}$ is traversed in clockwise order, and we know that \tilde{A}_{i+1} must be supported to the left. Therefore, $\tilde{A}_{i+2} \in N_L(\tilde{A}_{i+1}) \cup N_B(\tilde{A}_{i+1})$. Using the first part of the lemma, we conclude that, if $\tilde{A}_{i+2} \in N_L(\tilde{A}_{i+1})$ then $\tilde{A}_{i+2} = \tilde{A}_1$ or $\tilde{A}_{i+1} = \tilde{A}_1$, or if $\tilde{A}_{i+2} \in N_B(\tilde{A}_{i+1})$ then $\tilde{A}_{i+1} = \tilde{A}_1$.

The last lemma implies that either $\tilde{A}_{i+1} \in N_B(\tilde{A}_i)$ or $\tilde{A}_{i+1} \in N_R(\tilde{A}_i)$ holds for all $i = 2, \dots, k-2$; see Fig. 2. The next lemma shows that there are only two possible arrangements of the squares \tilde{A}_{k-1} and \tilde{A}_k :

Lemma 3 Either $\tilde{A}_k \in N_R(\tilde{A}_{k-1})$ or $\tilde{A}_k \in N_T(\tilde{A}_{k-1})$.

Proof We traverse ∂H_h from P in clockwise order. From the definition of \overline{PQ} and Lemma 1 we know that $P_{k,1}$ is a point on $L_{\tilde{A}_k}$. If $P_{k-1,k} \in L_{\tilde{A}_k}$, then $\tilde{A}_k \in N_R(\tilde{A}_{k-1})$;

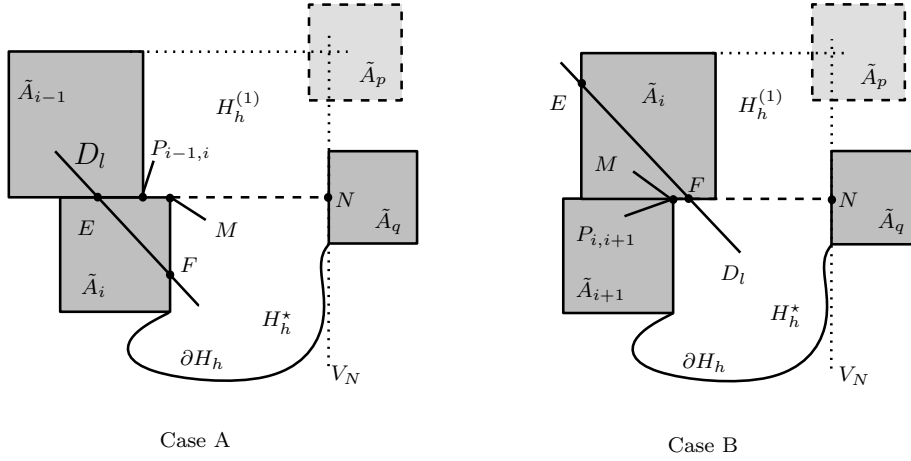


Fig. 4 D_l can intersect \tilde{A}_i (for the second time) in two different ways: on the right side or on the bottom side. In Case A, the square \tilde{A}_{i-1} is on top of \tilde{A}_i ; in Case B, \tilde{A}_i is on top of \tilde{A}_{i+1} .

if $P_{k-1,k} \in B_{\tilde{A}_k}$, then $\tilde{A}_k \in N_T(\tilde{A}_{k-1})$. In any other case \tilde{A}_k does not have a bottom neighbor.

Following the distinction described in the lemma, we say that a **hole** is of **Type I** if $\tilde{A}_k \in N_R(\tilde{A}_{k-1})$, and of **Type II** if $\tilde{A}_k \in N_T(\tilde{A}_{k-1})$; see Fig. 5.

2.2 Splitting Holes

Let H_h be a hole whose boundary does not touch the boundary of the strip, *i.e.*, the hole is completely enclosed by squares. We define two lines that are essential for the computation of an upper bound for the area of a hole, H_h : The *left diagonal*, D_l^h , is defined as the straight line with slope -1 starting in $P_{2,3}$ if $P_{2,3} \in R_{\tilde{A}_2}$ or, otherwise, in the lower right corner of \tilde{A}_2 ; see Fig. 5. We denote the point where D_l^h starts by P' . The *right diagonal*, D_r^h , is defined as the line with slope 1 starting in $P_{k-1,k}$ if $\tilde{A}_k \in N_R(\tilde{A}_{k-1})$ (Type I) or in $P_{k-2,k-1}$, otherwise (Type II). Note that $P_{k-2,k-1}$ lies on $L_{\tilde{A}_{k-1}}$ because otherwise, there would not be a left neighbor of \tilde{A}_{k-1} . We denote the point where D_r^h starts by Q' . If h is clear or does not matter we omit the superscript.

Lemma 4 *Let H_h be a hole and D_r its right diagonal. Then, $D_r \cap H_h^\circ = \emptyset$.*

Proof Consider the left sequence, $\mathcal{L}_{\tilde{A}_k} = (\tilde{A}_k = \alpha_1, \alpha_2, \dots)$ or $\mathcal{L}_{\tilde{A}_{k-1}} = (\tilde{A}_{k-1} = \alpha_1, \alpha_2, \dots)$, for H_h being of Type I or II, respectively. It is easy to show by induction that the upper left corners of the α_i 's lie above D_r : If D_r intersects $\partial\alpha_i$ at all, the first intersection is on R_{α_i} , the second on B_{α_i} . Thus, at least the skyline separates D_r and H_h .

If Lemma 4 would also hold for D_l , we could use the polygon formed by D_l , D_r , and the part of the boundary of H_h between Q' and P' to bound the area of H_h , but—unfortunately—it does not.

Let F be the first nontrivial intersection point of ∂H_h and D_l , while traversing ∂H_h in counterclockwise order, starting in P . F is on the boundary of a square, \tilde{A}_i . Let E be the other intersection point of D_l and $\partial \tilde{A}_i$.

It is a simple observation that if D_l intersects a square, \tilde{A}_i , in a nontrivial way, *i.e.*, in two different points, E and F , then either $F \in R_{\tilde{A}_i}$ and $E \in T_{\tilde{A}_i}$ or $F \in B_{\tilde{A}_i}$ and $E \in L_{\tilde{A}_i}$. To break ties, we define that an intersection in the lower right corner of \tilde{A}_i belongs to $B_{\tilde{A}_i}$. Now, we split our hole, H_h , into two new holes, $H_h^{(1)}$ and H_h^* . We consider two cases (see Fig. 4):

- Case A: $F \in R_{\tilde{A}_i} \setminus B_{\tilde{A}_i}$
- Case B: $F \in B_{\tilde{A}_i}$

In Case A, we define $\tilde{A}_{\text{up}} := \tilde{A}_{i-1}$ and $\tilde{A}_{\text{low}} := \tilde{A}_i$, in Case B $\tilde{A}_{\text{up}} := \tilde{A}_i$ and $\tilde{A}_{\text{low}} := \tilde{A}_{i+1}$. Observe the horizontal ray that emanates from the upper right corner of \tilde{A}_{low} to the right: This ray is subdivided into supported and unsupported sections. Let $U = \overline{MN}$ be the leftmost unsupported section with left endpoint M and right endpoint N ; see Fig. 4. Now, we split H_h into two parts, H_h^* below \overline{MN} and $H_h^{(1)} := H_h \setminus H_h^*$.

We split $H_h^{(1)}$ into $H_h^{(2)}$ and H_h^{**} etc., until there is no further intersection between the boundary of $H_h^{(z)}$ and D_l^h . Because there is a finite number of intersections, this process will eventually terminate. In the following, we show that $H_h^{(1)}$ and H_h^* are indeed two separate holes, and that H_h^* has the same properties as an original one, *i.e.*, it is a hole of Type I or II. Thus, we can analyze H_h^* using the same technique, *i.e.*, we may split H_h^* w.r.t. its left diagonal. We need some lemmas for this proof:

Lemma 5 *Using the above notation we have $\tilde{A}_{\text{low}} \in N_B(\tilde{A}_{\text{up}})$.*

Proof We consider Case A and Case B separately, starting with Case A. We traverse ∂H_h from F in clockwise order. By Lemma 1, \tilde{A}_{i-1} is the next square that we reach; see Fig. 4. Because F is the first intersection, $P_{i-1,i}$ lies between F and E . Thus, either $P_{i-1,i} \in T_{\tilde{A}_i}$ or $P_{i-1,i} \in R_{\tilde{A}_i}$ holds. With Lemma 2, the latter implies either $\tilde{A}_{i-1} = \tilde{A}_1$ or $\tilde{A}_i = \tilde{A}_1$. Because $b_{\tilde{A}_{i-1}} > b_{\tilde{A}_i}$ holds, only $\tilde{A}_{i-1} = \tilde{A}_1$ is possible, and therefore, $\tilde{A}_i = \tilde{A}_2$. D_l intersects \tilde{A}_2 in the lower left corner—which is not included in this case—or in $P_{2,3}$. However, $P_{2,3} \in R_{\tilde{A}_2}$ cannot be an intersection, because this would imply $\tilde{A}_3 \in N_R(\tilde{A}_2)$. Thus, only $P_{i-1,i} \in T_{\tilde{A}_i}$ is possible.

In Case B, we traverse ∂H_h from F in counterclockwise order, and \tilde{A}_{i+1} is the next square that we reach. Because F is the first intersection, it follows that $P_{i,i+1}$ lies on $\partial \tilde{A}_i$ between F and E in clockwise order; see Fig. 4. Thus, $\tilde{A}_{i+1} \in N_B(\tilde{A}_i)$ or $\tilde{A}_{i+1} \in N_L(\tilde{A}_i)$ holds. If $\tilde{A}_{i+1} \in N_L(\tilde{A}_i)$, we have $P_{i,i+1} \in L_{\tilde{A}_i}$. If we move from $P_{i,i+1}$ to F on $\partial \tilde{A}_i$, we move in clockwise order on ∂H_h . If we reach $P_{i-1,i}$ before F , the square, \tilde{A}_{i-1} , is between $P_{i,i+1}$ and F . The points, $P_{i,i+1}$ and F , are on ∂H_h , and thus, $\partial H_h \cap \tilde{A}_i$ is disconnected, which contradicts Lemma 1. Thus, we reach F before $P_{i-1,i}$. Moreover, \tilde{A}_i must have a bottom neighbor, and therefore, $P_{i-1,i} \in B_{\tilde{A}_i}^\circ$. By Lemma 2, we have $\tilde{A}_i = \tilde{A}_1$ or $\tilde{A}_{i+1} = \tilde{A}_1$. Both cases contradict the fact that D_l intersects neither \tilde{A}_2 in the lower right corner nor \tilde{A}_1 . Altogether, $P_{i,i+1}$ must be on $B_{\tilde{A}_i}$ to the left of F .

The last lemma states that in both cases, there are two squares for which one is indeed placed on top of the other.

Lemma 6 M is the upper right corner of \tilde{A}_{low} .

Proof Case A: We know $F \in R_{\tilde{A}_i}$ and $P_{i-1,i} \in T_{\tilde{A}_i}$. By Lemma 1, the upper right corner, M' , of \tilde{A}_i belongs to ∂H_h . Because F does not coincide with M' (degenerate intersection), $\overline{FM'}$ is a vertical line of positive length. Hence, M' is the beginning of an unsupported section of the horizontal ray emanating from M' to the right. Thus, the first unsupported section starts in M' ; that is, $M = M'$. A similar argument holds in Case B.

To ensure that H_h^* is well defined, we show that it has a closed boundary. Obviously, \overline{MN} and the part of ∂H_h counterclockwise from M to N forms a closed curve. We place an imaginary copy of \tilde{A}_{up} on \overline{MN} , such that the lower right corner is placed in N . We call the copy the *virtual lid*, denoted by \tilde{A}'_{up} . We show that $\overline{MN} < \tilde{a}_{\text{up}}$ holds, where \tilde{a}_{up} denotes the side length of \tilde{A}_{up} . Thus, \overline{MN} is completely covered by the virtual copy of \tilde{A}_{up} , and in turn, we can choose the virtual block as a new lid for H_h^* .

Lemma 7 With the above notation we have $\overline{MN} < \tilde{a}_{\text{up}}$.

Proof We show that at the time \tilde{A}_{up} is packed by BottomLeft, it can be moved to the right along \overline{MN} , such that the lower right corner coincides with N . Since \overline{MN} is unsupported, $\overline{MN} \geq \tilde{a}_{\text{up}}$ implies that there would have been a position for \tilde{A}_{up} that is closer to the bottom of S than its current position.

Let V_N be the vertical line passing through the point N , and let v_N be its x -coordinate. Assume that there is a square, \tilde{A}_p , that prevents \tilde{A}_{up} from being moved. Then, \tilde{A}_p fulfills $l_{\tilde{A}_p} < v_N$ and $b_{\tilde{A}_p} < t_{\tilde{A}_{\text{up}}}$ (*); see Fig. 4. Now, consider the sequence $\mathcal{L}_{\tilde{A}_p}$, and note that all squares in $\mathcal{L}_{\tilde{A}_p}$ are placed before \tilde{A}_{up} . From (*) we conclude that the skyline, $\mathcal{S}_{\tilde{A}_p}$, may intersect the horizontal line passing through $T_{\tilde{A}_{\text{low}}}$ only to the left of v_N . If the skyline intersects or touches in \overline{MN} , we have a contradiction to the choice of M and N as endpoints of the first unsupported section. An intersection between M and $P_{\text{up,low}}$ is not possible, because this part completely belongs to $T_{\tilde{A}_{\text{low}}}$. Therefore, $\mathcal{S}_{\tilde{A}_p}$ either intersects the horizontal line to the left of $P_{\text{up,low}}$ or it reaches L_S before. This implies that \tilde{A}_{up} must pass \tilde{A}_p on the right side and at the bottom side to get to its final position. In particular, $b_{\tilde{A}_p} < t_{\tilde{A}_{\text{up}}}$ implies that \tilde{A}_{up} 's path must go upwards to reach its final position; such a path contradicts the choice of BottomLeft.

Using the preceding lemmas, we can prove the following:

Corollary 1 Let H_h^* and \tilde{A}'_{up} be defined as above. H_h^* is a hole of Type I or Type II with virtual lid \tilde{A}'_{up} .

Proof H_h^* has a closed boundary, and there is at least a small area below \overline{MN} in which no squares are placed. Hence, H_h^* is a hole. Using the arguments that the interior of \overline{MN} is unsupported and that N is supported and lies on $L_{\tilde{A}_q}$, for some $1 \leq q \leq k$, we conclude that there is a vertical line of positive length below N on $\partial \tilde{A}_q$ that belongs to ∂H_h . If we move from N on $\partial \tilde{A}_q$ in counterclockwise order, we move on ∂H_h in clockwise order and reach \tilde{A}_{q-1} next. If $P_{q-1,q} \in L_{\tilde{A}_q}$, then H_h^* is of Type I. If $P_{q-1,q} \in B_{\tilde{A}_q}$, then it is of Type II. $P_{q-1,q} \notin L_{\tilde{A}_q} \cup B_{\tilde{A}_q}$ yields a contradiction, because in this case there is no bottom neighbor for \tilde{A}_q . \tilde{A}'_{up} is the unique lid by the existence of the sequences $\mathcal{B}_{\tilde{A}_q}$ and $\mathcal{B}_{\tilde{A}_{\text{low}}}$.

Note that the preceding lemmas also hold for the holes $H_h^{(\cdots)}$, H_h^{**} , H_h^{***} , and so on.

Lemma 8 *For every square, A_i , there is at most one copy of A_i .*

Proof A square, A_i , is used as a virtual lid, only if its lower right corner is on the boundary of the hole that is split. Because its corner can be on the boundary of at most one hole, there is only one hole with virtual lid A_i .

2.3 Computing the Area of a Hole

In this section we show how to compute the area of a hole. In the preceding section we eliminated all intersections of D_l^h with the boundary of the hole, $H_h^{(z)}$, by splitting the hole. Thus, we assume that we have a set of holes, \hat{H}_h , $h = 1, \dots, s^*$, that fulfill $\partial\hat{H}_h \cap D_l^h = \emptyset$ and have either a non-virtual or a virtual lid.

Our aim is to bound $|\hat{H}_h|$ by the areas of the squares that contribute to $\partial\hat{H}_h$. A square, A_i , may contribute to more than one hole. Therefore, it is too expensive to use its total area, a_i^2 , in the bound for a single hole. Instead, we charge only fractions of a_i^2 per hole. Moreover, we charge every edge of A_i separately. By Lemma 1, $\partial\hat{H}_h \cap \partial A_i$ is connected. In particular, every side of A_i contributes at most one (connected) line segment to $\partial\hat{H}_h$. For the left (bottom, right) side of a square, A_i , we denote the length of the line segment contributed to $\partial\hat{H}_h$ by λ_i^h (β_i^h , ρ_i^h ; respectively). If a side of a square does not contribute to a hole, the corresponding length of the line segment is defined to be zero.

Let $c_{h,i}^{\{\lambda,\beta,\rho\}}$ be appropriate coefficients, such that the area of a hole can be charged against the area of the adjacent squares, *i.e.*,

$$|\hat{H}_h| \leq \sum_{i=1}^{n+1} c_{h,i}^\lambda (\lambda_i^h)^2 + c_{h,i}^\beta (\beta_i^h)^2 + c_{h,i}^\rho (\rho_i^h)^2.$$

As each point on ∂A_i is—obviously—on the boundary of at most one hole, the line segments are pairwise disjoint. Thus, for the left side of A_i , the two squares inside A_i induced by the line segments, λ_i^h and λ_i^g , of two different holes, \hat{H}_h and \hat{H}_g , do not overlap. Therefore, we obtain

$$\sum_{h=1}^{s^*} c_{h,i}^\lambda \cdot (\lambda_i^h)^2 \leq c_i^\lambda \cdot a_i^2,$$

where c_i^λ is the maximum of the $c_{h,i}^\lambda$'s taken over all holes \hat{H}_h . We call c_i^λ the *charge* of L_{A_i} and define c_i^β and c_i^ρ analogously.

We use virtual copies of some squares as lids. However, for every square, A_i , there is at most one copy, A'_i . We denote the line segments and charges corresponding to A'_i by $\lambda_{i'}^h$, $c_{h,i'}^\lambda$, and so on. Taking the charges to the copy into account, the *total charge* of A_i is given by

$$c_i = c_i^\lambda + c_i^\beta + c_i^\rho + c_{i'}^\lambda + c_{i'}^\beta + c_{i'}^\rho.$$

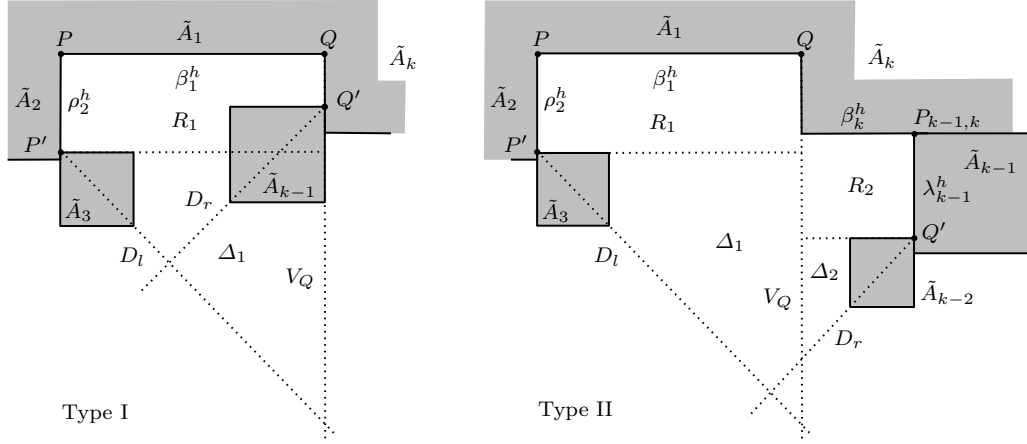


Fig. 5 Holes of Type I and II with their left and right diagonals.

Altogether, we bound the total area of the holes by

$$\sum_{h=1}^{s^*} |\hat{H}_h| \leq \sum_{i=1}^{n+1} c_i \cdot a_i^2 \leq \sum_{i=1}^{n+1} c \cdot a_i^2,$$

where $c = \max_{i=1, \dots, n} \{c_i\}$. In the following, we want to find an upper bound on c .

Holes with a Non-Virtual Lid We know that each hole is either of Type I or II. Moreover, we removed all intersections of \hat{H}_h with its diagonal, D_l^h . Therefore, \hat{H}_h lies completely inside the polygon formed by D_l^h , D_r^h , and the part of $\partial \hat{H}_h$ that is clockwise between P' and Q' ; see Fig. 5.

If \hat{H}_h is of Type I, then we consider the rectangle, R_1 , of area $\rho_2^h \cdot \beta_1^h$ induced by the points P , P' , and Q . Moreover, let Δ_1 be the triangle below R_1 formed by the bottom side of R_1 , D_l^h , and the vertical line, V_Q , passing through Q ; see Fig. 5. We obtain:

Lemma 9 *Let \hat{H}_h be a hole of Type I. Then,*

$$|\hat{H}_h| \leq (\beta_1^h)^2 + \frac{1}{2}(\rho_2^h)^2.$$

Proof Obviously, $|\hat{H}_h| \leq |R_1| + |\Delta_1|$. As D_l^h has slope -1 , we get $|\Delta_1| = \frac{1}{2}(\beta_1^h)^2$. Moreover, we have $|R_1| = \rho_2^h \cdot \beta_1^h \leq \frac{1}{2}(\rho_2^h)^2 + \frac{1}{2}(\beta_1^h)^2$. Altogether, we get the stated bound.

Thus, we charge the bottom side¹ of \tilde{A}_1 with 1 and the right side of \tilde{A}_2 with $\frac{1}{2}$. In this case, we get $c_{h,1}^\beta = 1$ and $c_{h,2}^\rho = \frac{1}{2}$.

¹ The charge to the bottom of \tilde{A}_1 can be reduced to $\frac{3}{4}$ by considering the larger one of the rectangles, R_1 and the one induced by Q , Q' , and P , as well as the triangle below the larger rectangle formed by D_l^h and D_r^h . However, this does not lead to a better competitive ratio, because these costs are already dominated by the cost for holes of Type II.

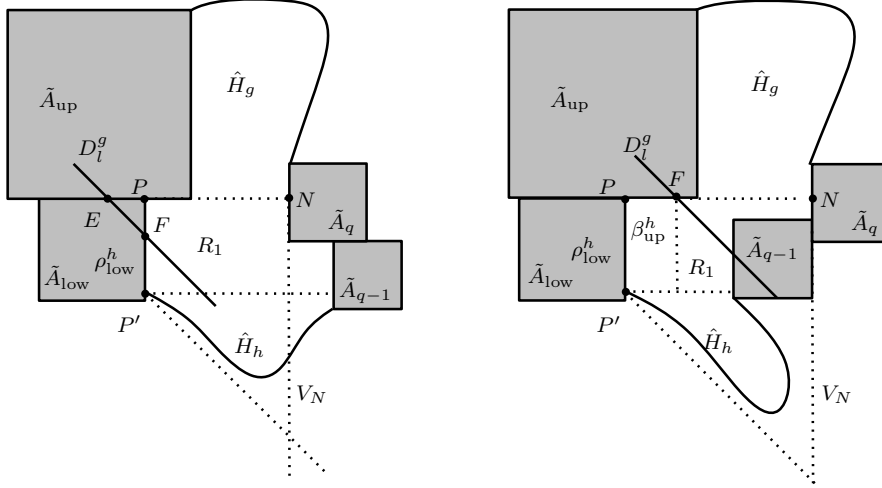


Fig. 6 The holes \hat{H}_g and \hat{H}_h and the rectangle R_1 which is divided into two parts by D_l^g . The upper part is already included in the bound for \hat{H}_g . The lower part is charged completely to $R_{\tilde{A}_{\text{low}}}$ and $B_{\tilde{A}'_{\text{up}}}$. Here P and P' are defined w.r.t. \hat{H}_h .

If \hat{H}_h is of Type II, we define R_1 and Δ_1 in the same way. In addition, R_2 is the rectangle of area $\beta_k^h \cdot \lambda_{k-1}^h$ induced by the points Q' and $P_{k-1,k}$ as well as the part of $B_{\tilde{A}_k}$ that belongs to $\partial\hat{H}_h$. Let Δ_2 be the triangle below R_2 , induced by the bottom side of R_2 , D_r^h , and V_Q . Using similar arguments as in the preceding lemma, we get:

Corollary 2 *Let \hat{H}_h be a hole of Type II. Then,*

$$|\hat{H}_h| \leq (\beta_1^h)^2 + (\beta_k^h)^2 + \frac{1}{2}(\rho_2^h)^2 + \frac{1}{2}(\lambda_{k-1}^h)^2.$$

We obtain the charges $c_{h,1}^\beta = 1$, $c_{h,2}^\rho = \frac{1}{2}$, $c_{h,k}^\beta = 1$ and $c_{h,k-1}^\lambda = \frac{1}{2}$. Thus, we have a maximum total charge of 2 (bottom: 1, left: 1/2, and right: 1/2) for a square, so far.

Holes with a Virtual Lid Next we consider a hole, \hat{H}_h , with a virtual lid. Let \hat{H}_g be the hole immediately above \hat{H}_h , i.e., \hat{H}_h was created by removing the diagonal-boundary intersections in \hat{H}_g . Corresponding to Lemma 5, let \tilde{A}_{up} be the square whose copy becomes a new lid, while \tilde{A}'_{up} is the copy. The bottom neighbor of \tilde{A}_{up} is denoted by \tilde{A}_{low} . We show that \tilde{A}'_{up} increases the total charge of \tilde{A}_{up} not above 2.5. Recall that \hat{H}_h is a hole of Type I or II by Corollary 1.

If \tilde{A}_{up} does not exceed \tilde{A}_{low} to the left, it cannot serve as a lid for any other hole; see Fig. 6. Hence, the charge of the bottom side of \tilde{A}_{up} is zero; by Corollary 1, Lemma 9, and Corollary 2 we obtain a charge of at most 1 to the bottom side of \tilde{A}'_{up} . Thus, we get a total charge of 1 to \tilde{A}_{up} . For an easier summation of the charges at the end, we transfer the charge from the bottom side of \tilde{A}'_{up} to the bottom side of \tilde{A}_{up} .

If it exceeds \tilde{A}_{low} to the left, we know that the part $B_{\tilde{A}_{\text{up}}} \cap T_{\tilde{A}_{\text{low}}}$ of $B_{\tilde{A}_{\text{up}}}$ is not charged by any other hole, because it does not belong to the boundary of a hole, and the lid is defined uniquely.

We define points, P and P' , for \hat{H}_h in the same way as in the preceding section. Independent of \hat{H}_h 's type, \tilde{A}'_{up} would get charged only for the rectangle R_1 induced by P , P' , and N , as well as for the triangle below R_1 if we would use Lemma 9 and Corollary 2.

Next we show that we do not have to charge \tilde{A}'_{up} for R_1 at all, because the part of R_1 that is above D_l^g is already included in the bound for \hat{H}_g , and the remaining part can be charged to $B_{\tilde{A}_{\text{up}}}$ and $R_{\tilde{A}_{\text{low}}}$. \tilde{A}'_{up} will get charged only $\frac{1}{2}$ for the triangle.

D_l^g splits R_1 into a part that is above this line and a part that is below this line. The latter part of R_1 is not included in the bound for \hat{H}_g . Let F be the intersection of $\partial\hat{H}_g$ and D_l^g that caused the creation of \hat{H}_h . If $F \in R_{\tilde{A}_{\text{low}}}$, then this part is at most $\frac{1}{2}(\rho_{\text{low}}^h)^2$, where ρ_{low}^h is the length of $\overline{P'F}$. We charge $\frac{1}{2}$ to $R_{\tilde{A}_{\text{low}}}$. If $F \in B_{\tilde{A}_{\text{up}}}$, then the part of R_1 below D_l^g can be split into a rectangular part of area $\rho_{\text{low}}^h \cdot \beta_{\text{up}}^h$, and a triangular part of area $\frac{1}{2}(\rho_{\text{low}}^h)^2$; see Fig. 6. Here β_{up}^h is the length of \overline{PF} . The cost of the triangular part is charged to $R_{\tilde{A}_{\text{low}}}$. Note that $B_{\tilde{A}_{\text{up}}}$ exceeds \tilde{A}_{low} to the left and to the right and that the part that exceeds \tilde{A}_{low} to the right is not charged. Moreover, ρ_{low}^h is not larger than $B_{\tilde{A}_{\text{up}}} \cap T_{\tilde{A}_{\text{low}}}$, i.e., the part of $B_{\tilde{A}_{\text{up}}}$ that was not charged before. Therefore, we can charge the rectangular part completely to $B_{\tilde{A}_{\text{up}}}$. Hence, \tilde{A}'_{up} is charged $\frac{1}{2}$ for the triangle below R_1 , and \tilde{A}_{up} is charged at most 2.5 in total.

Holes Containing Parts of ∂S So far we did not consider holes whose boundary touches ∂S . We show in this section that these holes are just special cases of the ones discussed in the preceding sections.

Because the top side of a square never gets charged for a hole, it does not matter whether a part of B_S belongs to the boundary. Moreover, for any hole, \hat{H}_h , either L_S or R_S can be a part of $\partial\hat{H}_h$, because otherwise there exists a curve with one endpoint on L_S and the other endpoint on R_S , with the property that this curve lies completely inside of \hat{H}_h . This contradicts the existence of the bottom sequence of a square lying above the curve.

For a hole \hat{H}_h with L_S contributing to $\partial\hat{H}_h$, we can use the same arguments as in the proof for Lemma 1 to show that $L_S \cap \partial\hat{H}_h$ is a single line segment. Let P be the topmost point of this line segment and \tilde{A}_1 be the square containing P . \tilde{A}_1 must have a bottom neighbor, \tilde{A}_k , and \tilde{A}_k must have a left neighbor, \tilde{A}_{k-1} , we get $P_{k,1} \in B_{\tilde{A}_1}$ and $P_{k-1,k} \in L_{\tilde{A}_k}$, respectively. We define the right diagonal, D_r , and the point Q' as above and conclude that \hat{H}_h lies completely inside the polygon formed by $L_S \cap \partial\hat{H}_h$, D_r , and the part of $\partial\hat{H}_h$ that is between P and Q' in clockwise order. We split this polygon into a rectangle and a triangle in order to obtain charges of 1 to $B_{\tilde{A}_1}$ and $\frac{1}{2}$ to $L_{\tilde{A}_k}$.

Now, consider a hole where a part of R_S belongs to $\partial\hat{H}_h$. We denote the topmost point on $R_S \cap \partial\hat{H}_h$ by Q , and the square containing Q by \tilde{A}_1 . We number the squares in counterclockwise order and define the left diagonal, D_l , as above. Now we consider the intersections of D_l and eliminate them by creating new holes. After this, the modified hole $\hat{H}_h^{(z)}$ can be viewed as a hole of Type II, for which the part on the right side of V_Q has been cut off; compare Corollary 2. We obtain charges of 1 to $B_{\tilde{A}_1}$ and $\frac{1}{2}$ to $R_{\tilde{A}_2}$. For the copy of a square we get a charge of $\frac{1}{2}$ to the bottom side.

	Non-virtual Lid					Virtual Lid				Total
	Type I	Type II	L_S	R_S	Max.	Type I	Type II	R_S	Max.	
Left Side	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	$\frac{1}{2}$
Bottom Side	1	1	1	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1.5
Right Side	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	$\frac{1}{2}$
Total					2				$\frac{1}{2}$	2.5

Table 1 The charges to the different sides of a single square. Summing up the charges to the different sides, we conclude that every square gets a total charge of at most 2.5.

2.4 Summing up the Charges

Altogether, we have the charges from Table 1. The charges depend on the type of the adjacent hole (Type I, II, touching or not touching the strip's boundary), but the maximal charge dominates the other ones. Moreover, the square may also serve as a virtual lid. The maximal charges from a hole with non-virtual lid and those from a hole with virtual lid sum up to a total charge of 2.5 per square. This proves our claim from the beginning:

$$\sum_{h=1}^s |H_h| \leq 2.5 \cdot \sum_{i=1}^{n+1} a_i^2.$$

3 The Strategy *SlotAlgorithm*

In this section we analyze a different strategy for the strip packing problem with Tetris and gravity constraint. This strategy provides more structure on the generated packing, which allows us to prove an upper bound of 2.6154 on the asymptotic competitive ratio.

3.1 The Algorithm

Consider two vertical lines of infinite length going upwards from the bottom side of S and parallel to the left and the right side of S . We call the area between these lines a *slot*, the lines the *left boundary* and the *right boundary* of the slot, and the distance between the lines the *width* of the slot.

Now, our strategy *SlotAlgorithm* works as follows: We divide the strip S of width 1 into slots of different widths; for every $j = 0, 1, 2 \dots$, we create 2^j slots of width 2^{-j} side by side, *i.e.*, we divide S into one slot of width 1, two slots of width $1/2$, four slots of width $1/4$, and so on. Note that a slot of width 2^{-j} contains 2 slots of width 2^{-j-1} ; see Fig. 7.

For every square A_i , we round the side length a_i to the smallest number 2^{-k_i} that is larger than or equal to a_i . Among all slots of width 2^{-k_i} , we place A_i in the one that allows A_i to be placed as near to the bottom of S as possible, by moving A_i down along the left boundary of the chosen slot until another square is reached. The

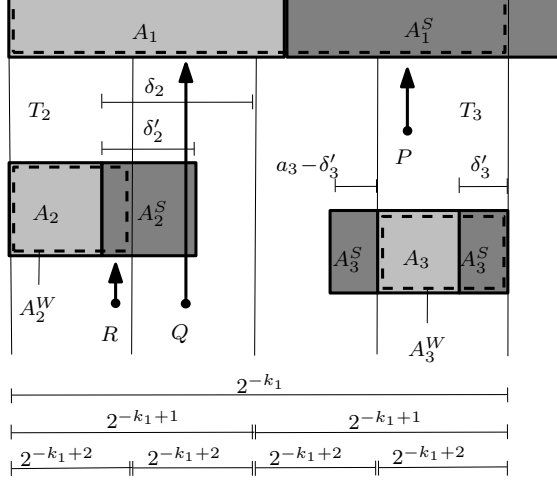


Fig. 7 Squares A_i , $i = 1, 2, 3$, with their shadows A_i^S and their widening A_i^W . δ'_2 is equal to a_2 and δ'_3 is equal to δ_3 . The points P and Q are charged to A_1 . R is not charged to A_1 , but to A_2 .

algorithm clearly satisfies the Tetris and the gravity constraints, and next we show that the produced height is at most 2.6154 times the height of an optimal packing.

3.2 Analysis

Let A_i be a square placed by the SlotAlgorithm in a slot T_i of width 2^{-k_i} . If $a_i \leq \frac{1}{2}$, we define δ_i as the distance between the right side of A_i and the right boundary of the slot of width 2^{-k_i+1} that contains A_i , and we define $\delta'_i = \min\{a_i, \delta_i\}$. We call the area obtained by enlarging A_i by δ'_i to the right and by $a_i - \delta'_i$ to the left (without A_i itself) the *shadow* of A_i and denote it by A_i^S . Thus, A_i^S is an area of the same size as A_i and lies completely inside a slot of twice the width of A_i 's slot. If $a_i \geq \frac{1}{2}$, we enlarge A_i only to the right side and call this area the shadow. Moreover, we define the *widening* of A_i as $A_i^W = (A_i \cup A_i^S) \cap T_i$; see Fig. 7.

Now, consider a point P in S that is not inside an A_j^W for any square A_j . We charge P to the square, A_i , if A_i^W is the first widening that intersects the vertical line going upwards from P . We denote by F_{A_i} the set of all points charged to A_i and by $|F_{A_i}|$ its area. For points lying on the left or the right boundary of a slot, we break ties arbitrarily. For the analysis, we place a closing square, A_{n+1} , of side length 1 on top of the packing. Therefore, every point in the packing that does not lie inside an A_j^W is charged to a square. Because A_i and A_i^S have the same area, we can bound the height of the packing produced by the SlotAlgorithm by

$$2 \cdot \sum_{i=1}^n a_i^2 + \sum_{i=1}^{n+1} |F_{A_i}|.$$

Theorem 2 *The SlotAlgorithm is (asymptotically) 2.6154-competitive.*

Proof The height of an optimal packing is at least $\sum_{i=1}^n a_i^2$, and therefore, it suffices to show that $|F_{A_i}| \leq 0.6154 \cdot a_i^2$ holds for every square A_i . We construct for every A_i a sequence of squares $\tilde{A}_1^i, \tilde{A}_2^i, \dots, \tilde{A}_m^i$ with $\tilde{A}_1^i = A_i$ (to ease notation, we omit the superscript i in the following). We denote by E_j the extension of the bottom side of \tilde{A}_j to the left and to the right; see Fig. 8.

We will show that by an appropriate choice of the sequence, we can bound the area of the part of $F_{\tilde{A}_1}$ that lies between a consecutive pair of extensions, E_j and E_{j+1} , in terms of \tilde{A}_{j+1} and the slot width. From this we will derive the upper bound on the area of $F_{\tilde{A}_1}$. We assume throughout the proof that the square \tilde{A}_j , $j \geq 1$, is placed in a slot, T_j , of width 2^{-k_j} . Note that $F_{\tilde{A}_1}$ is completely contained in T_1 .

A slot is called *active* (with respect to E_j and \tilde{A}_1) if there is a point in the slot that lies below E_j and that is charged to \tilde{A}_1 and *nonactive* otherwise. If it is clear from the context we leave out the \tilde{A}_1 .

The sequence of squares is chosen as follows: \tilde{A}_1 is the first square and the next square, \tilde{A}_{j+1} , $j = 1, \dots, m-1$, is chosen as the smallest one that intersects or touches E_j in an active slot (w.r.t. E_j and \tilde{A}_1) of width 2^{-k_j} and that is not equal to \tilde{A}_j ; see Fig. 8. The sequence ends if all slots are nonactive w.r.t. to an extension E_m . We prove each of the following claims by induction:

Claim A: \tilde{A}_{j+1} exists for $j+1 \leq m$ and $\tilde{a}_{j+1} \leq 2^{-k_j-1}$ for $j+1 \leq m$, i.e., the sequence exists and its elements have decreasing side length.

Claim B: The number of active slots (w.r.t. E_j) of width 2^{-k_j} is at most

$$\begin{aligned} & 1, \text{ for } j = 1 \text{ and} \\ & \prod_{i=2}^j \left(\frac{1}{2^{k_{i-1}}} 2^{k_i} - 1 \right), \text{ for } j \geq 2. \end{aligned}$$

Claim C: The area of the part of $F_{\tilde{A}_1}$ that lies in an active slot of width 2^{-k_j} between E_j and E_{j+1} is at most $2^{-k_j} \tilde{a}_{j+1} - 2\tilde{a}_{j+1}^2$.

Proof of Claim A: If \tilde{A}_1 is placed on the bottom of S , $F_{\tilde{A}_1}$ has size 0 and \tilde{A}_1 is the last element of the sequence. Otherwise, the square \tilde{A}_1 has at least one bottom neighbor, which is a candidate for the choice of \tilde{A}_2 .

Now suppose for a contradiction that there is no candidate for the choice of the $(j+1)$ th element. Let T' be an active slot in T_1 (w.r.t. E_j) of width 2^{-k_j} where E_j is not intersected by a square in T' . If there is an ε such that for every point, $P \in (T' \cap E_j)$, there is a point, P' , at a distance ε below P which is charged to \tilde{A}_1 , we conclude that there would have been a better position for \tilde{A}_j . Hence, there is at least one point, Q , below E_j that is not charged to \tilde{A}_1 ; see Fig 9. Consider the bottom sequence (as defined in Section 2.1) of the square Q is charged to. This sequence must intersect E_j outside of T' (by the choice of T'). This implies that one of its elements must intersect the left or the right boundary of T' , and we can conclude that this square has at least the width of T' . This is because (by the algorithm) a square with rounded side length $2^{-\ell}$ cannot cross a slot's boundary of width larger than $2^{-\ell}$. In turn, a square with rounded side length larger than the width of T' completely covers T' , and T' cannot be active w.r.t. to E_j and \tilde{A}_1 . Thus, all points in T' below E_j are charged to this square; a contradiction. This proves that there is a candidate for the choice of \tilde{A}_{j+1} .

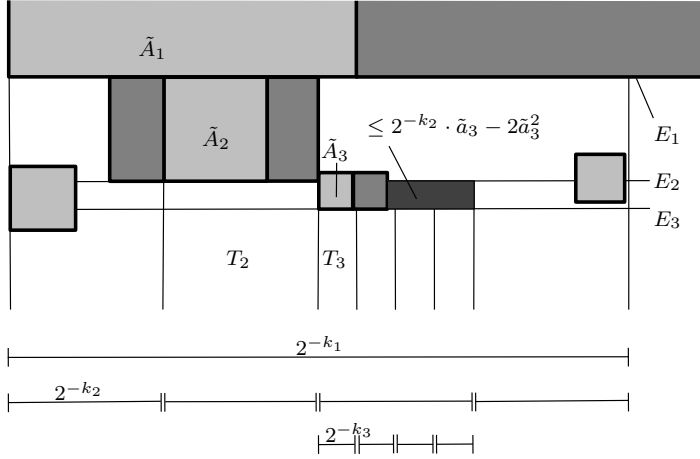


Fig. 8 The first three squares of the sequence (light gray) with their shadows (gray). In this example, \tilde{A}_2 is the smallest square that bounds \tilde{A}_1 from below. \tilde{A}_3 is the smallest one that intersects E_2 in an active slot (w.r.t. E_2) of width 2^{-k_2} . There has to be an intersection of E_2 and some square in every active slot because, otherwise, there would have been a better position for \tilde{A}_2 . T_2 is nonactive, (w.r.t. E_2) and of course, also w.r.t. all extension E_j , $j \geq 3$. The part of $F_{\tilde{A}_1}$ that lies between E_1 and E_2 has size $2^{-k_1}\tilde{a}_2 - 2\tilde{a}_2^2$.

Suppose $\tilde{a}_2 > 2^{-k_1-1}$. Then, \tilde{A}_2 was placed in a slot of width at least 2^{-k_1} . Thus, its widening has width at least 2^{-k_1} , and \tilde{A}_2 is a bottom neighbor of \tilde{A}_1 . Then, no point in T_1 , below E_1 , is charged to \tilde{A}_1 , and hence, T_1 is nonactive w.r.t. E_1 and \tilde{A}_1 . This implies, that \tilde{A}_2 does not belong to the sequence; a contradiction.

Because we chose \tilde{A}_{j+1} to be of minimal side length, $\tilde{a}_{j+1} \geq 2^{-k_j}$ would imply that all slots inside T are nonactive (w.r.t. E_j). Therefore, if \tilde{A}_{j+1} belongs to the sequence, $\tilde{a}_{j+1} \leq 2^{-k_j-1}$ must hold.

Proof of Claim B: Obviously, there is at most one active slot of width 2^{-k_1} ; see Fig. 8. By the induction hypothesis, there are at most

$$\left(\frac{1}{2^{k_1}}2^{k_2} - 1\right) \cdot \left(\frac{1}{2^{k_2}}2^{k_3} - 1\right) \cdot \dots \cdot \left(\frac{1}{2^{k_{j-2}}}2^{k_{j-1}} - 1\right)$$

active slots of width $2^{-k_{j-1}}$ (w.r.t. E_{j-1}). Each of these slots contains $2^{k_j-k_{j-1}}$ slots of width 2^{-k_j} , and in every active slot of width $2^{-k_{j-1}}$ at least one slot of width 2^{-k_j} is nonactive because we chose \tilde{A}_j to be of minimum side length. Hence, the number of active slots (w.r.t. E_j) is a factor of $\left(\frac{1}{2^{k_{j-1}}}2^{k_j} - 1\right)$ times larger than the number of active slots (w.r.t. E_{j-1}).

Proof of Claim C: The area of the part of $F_{\tilde{A}_1}$ that lies between E_1 and E_2 is at most $2^{-k_1}\tilde{a}_2 - 2\tilde{a}_2^2$; see Fig. 8. Note that we can subtract the area of \tilde{A}_2 twice, because \tilde{A}_2^S was defined to lie completely inside a slot of width $2^{-k_2+1} \leq 2^{-k_1}$ and is of same area as \tilde{A}_2 .

By the choice of \tilde{A}_{j+1} and because in every active slot of width 2^{-k_j} there is at least one square that intersects E_j (points below the widening of this square are not

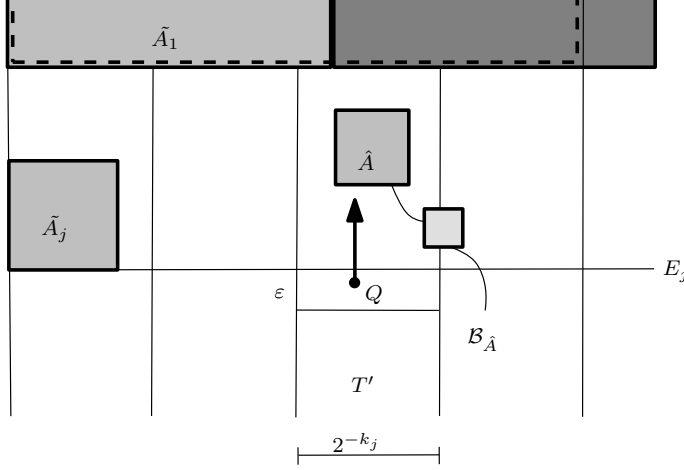


Fig. 9 If E_j is not intersected in an active slot of size 2^{-k_j} we obtain a contradiction: Either there is a position for A_j that is closer to the bottom of S or there is a square that makes E_j nonactive. \hat{A} is the square Q is charged to, $\mathcal{B}_{\hat{A}}$ its bottom sequence.

charged to \tilde{A}_1) we conclude that the area of $F_{\tilde{A}_1}$ between E_j and E_{j+1} is at most $2^{-k_j} \tilde{a}_{j+1} - 2\tilde{a}_{j+1}^2$, in every active slot of width 2^{-k_j} .

Altogether, we proved that the sequence is well defined and we calculated an upper bound on the number of active slots and an upper bound on the size of the part of $|F_{\tilde{A}_1}|$ that is contained in an active slot. Multiplying the number and the size yields an upper bound on $|F_{\tilde{A}_1}|$ of

$$|F_{\tilde{A}_1}| \leq \left(\frac{\tilde{a}_2}{2^{k_1}} - 2\tilde{a}_2^2\right) \cdot 1 + \sum_{j=2}^m \left(\frac{\tilde{a}_{j+1}}{2^{k_j}} - 2\tilde{a}_{j+1}^2\right) \prod_{i=2}^j \left(\frac{2^{k_i}}{2^{k_{i-1}}} - 1\right).$$

This expression is maximized if we choose $\tilde{a}_{i+1} = 1/2^{k_i+2}$, for $i = 1, \dots, m$, i.e., $k_i = k_1 + 2(i-1)$.

We get:

$$\begin{aligned}
|F_{\tilde{A}_1}| &\leq \frac{1}{2^{k_1+2}} \cdot \frac{1}{2^{k_1}} - 2 \cdot \left(\frac{1}{2^{k_1+2}} \right)^2 \\
&\quad + \sum_{j=2}^m \left[\frac{1}{2^{k_1+2(j-1)}} \cdot \frac{1}{2^{k_1+2j}} - 2 \left(\frac{1}{2^{k_1+2j}} \right)^2 \right] \prod_{i=1}^{j-1} \left(\frac{2^{k_1+2i}}{2^{k_1+2(i-1)}} - 1 \right) \\
&= \frac{1}{2^{k_1+3}} + \sum_{j=2}^m \left[\frac{1}{2^{2k_1+4j-2}} - \frac{1}{2^{2k_1+4j-1}} \right] \cdot 3^{j-1} \\
&= \frac{1}{2^{k_1+3}} + \sum_{j=2}^m \frac{3^{j-1}}{2^{2k_1+4j-1}} \\
&= \frac{1}{2^{k_1+3}} + \sum_{j=1}^{m-1} \frac{3^j}{2^{2k_1+4j+3}} \\
&\leq \sum_{j=0}^{\infty} \frac{3^j}{2^{2k_1+4j+3}}.
\end{aligned}$$

The fraction $|F_{\tilde{A}_1}|/\tilde{a}_1^2$ is maximized, if we choose \tilde{a}_1 as small as possible, *i.e.*, $\tilde{a}_1 = 2^{-k_1-1} + \varepsilon$. We conclude:

$$\frac{|F_{\tilde{A}_1}|}{\tilde{a}_1^2} \leq \sum_{j=0}^{\infty} \frac{2^{2k_1+2} \cdot 3^j}{2^{2k_1+4j+3}} = \sum_{j=0}^{\infty} \frac{3^j}{2^{4j+1}} = \frac{1}{2} \cdot \sum_{j=0}^{\infty} \left(\frac{3}{16} \right)^j = \frac{8}{13} = 0.6153...$$

Thus,

$$|F_{A_i}| \leq 0.6154 \cdot a_i^2.$$

4 Lower Bounds

The lower bound construction for online strip packing introduced by Galambos and Frenk [14] and later improved by van Vliet [24] relies on an integer programming formulation and its LP-relaxation for a specific bin packing instance. This formulation does not take into account that there has to be a collision free path to the final position of the item. Hence, it does not carry over to our setting.

The best asymptotic lower bound, we are aware of, is $\frac{5}{4}$. It is based on two sequences which are repeated iteratively. We denote by A_i^k , $k = 1, \dots, 5$ and $i = 1, 2, \dots$, the k -th square of the sequence in the i -th iteration, and we denote by H_i , $i = 1, 2, \dots$, the height of the packing after the i -th iteration; we define $H_0 = 0$.

The first two squares of each sequence have a side length of $\frac{1}{4}$, that is, $a_i^1 = a_i^2 = \frac{1}{4}$. Now, depending on the choice of the algorithm, the sequence continues with one of the following two possibilities (see Fig. 10):

Type I: If the algorithm packs the first two squares on top of each other, with the bottom side of the lower square at height H_{i-1} , the sequence continues with a square of side length $\frac{3}{4} + \varepsilon$, *i.e.*, $a_i^3 = \frac{3}{4} + \varepsilon$ and $a_i^4 = a_i^5 = 0$ (upper left picture in Fig. 10).

Type II: Otherwise, the sequence continues with a square of side length $\frac{1}{2} + \varepsilon$ and two squares of side length $\frac{1}{2}$, *i.e.*, $a_i^3 = \frac{1}{2} + \varepsilon$ and $a_i^4 = a_i^5 = \frac{1}{2}$ (lower left picture in Fig. 10).

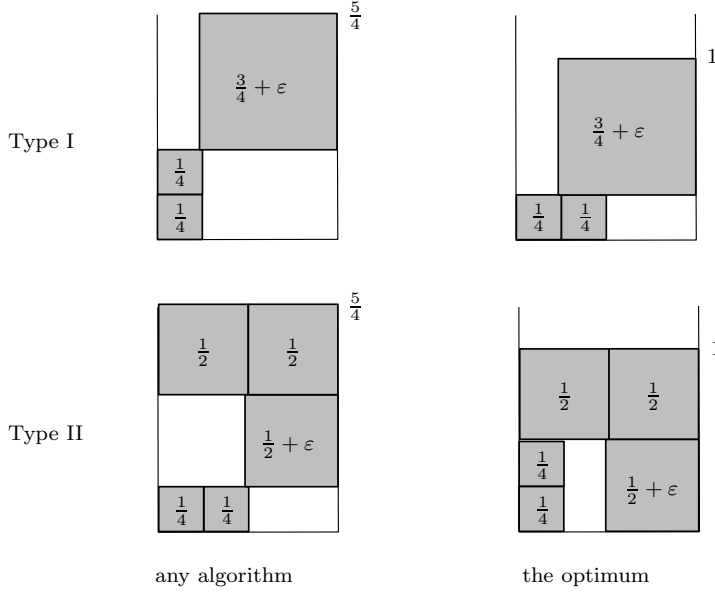


Fig. 10 The left column shows possible packings of any algorithm for one iteration. The right column contains optimal packings. The top row displays the first and the bottom row the second type of sequence.

Lemma 10 *The height of the packing produced by any algorithm increases in each iteration, on average, by at least $\frac{5}{4}$.*

Proof Consider the i -th iteration, $i = 1, 2, \dots$. If the sequence is of Type I, the statement is obviously true because the square of side length $\frac{3}{4} + \varepsilon$ cannot pass any of the squares of side length $\frac{1}{4}$ which are packed on top of each other; see Fig. 10.

If the sequence is of Type II, we need to consider the previous iteration. If there was no previous iteration, then we know that A_i^1 and A_i^2 are both placed on the bottom side of the strip. Because A_i^3 cannot be placed on the bottom side, and A_i^4 and A_i^5 cannot pass A_i^3 , we get an increase of at least $\frac{5}{4}$.

If the sequence in the previous iteration was of Type I, H_{i-1} is determined by the square of side length $\frac{3}{4} + \varepsilon$. Hence, A_i^1 and A_i^2 are both placed on top of this square and the same arguments hold.

If the sequence in the previous iteration was of Type II, then either A_{i-1}^4 and A_{i-1}^5 are packed next to each other or on top of each other. In the first case, we can use the same arguments as in the case where there was no previous iteration. In the second case, A_{i-1}^4 and A_{i-1}^5 are placed on top of each other and on top of A_{i-1}^3 , because they cannot pass a square with side length $\frac{1}{2} + \varepsilon$. This implies that, the last iteration contributed a height of at least $\frac{3}{2}$ to the height of the packing. No matter how the algorithm packs the squares from the current iteration (the first two squares might be placed at the same height or even deeper as the previous squares) it contributes a height of at least 1 to the packing. This proves an average increase of $\frac{5}{4}$ for both iterations.

Theorem 3 *There is no algorithm with asymptotic competitive ratio smaller than $\frac{5}{4}$ for the online strip packing problem with Tetris and gravity constraint.*

Proof The height of the packing produced by any algorithm increases by $\frac{5}{4}$ per iteration for the above instance (Lemma 10). The optimum can pack the squares belonging to one iteration always such that the height of the packing increases by at most 1; see the right column of Fig. 10.

5 Conclusion

There are instances consisting only of squares for which the algorithm of Azar and Epstein does not undercut its proven competitive factor of 4. Hence, this algorithm is tightly analyzed. We proved competitive ratios of 3.5 and 2.6154 for BottomLeft and the SlotAlgorithm, respectively. Hence, both algorithms outperform the one by Azar and Epstein if the input consists only of squares.

We do not know any instance for which BottomLeft produces a packing that is 3.5 times higher than an optimal packing. The best lower bound we know is $\frac{5}{4}$.

Moreover, we are not aware of an instance in which the SlotAlgorithm reaches its upper bound of 2.6154. The instance consisting of squares with side length $2^{-k} + \delta$, for large k and small δ , gives a lower bound of 2 on the competitive ratio.

Hence, there is still room for improvement: Our analysis might be improved or there may be more sophisticated algorithms for the strip packing problem with Tetris and gravity constraint.

At this point, the bottleneck in our analysis for BottomLeft is the case in which a square has large holes at the right, left, and bottom side and also serves as a virtual lid; see Fig. 2. This worst case can happen to only a few squares, but never to all of them. Thus, it might be possible to transfer charges between squares, which may yield a refined analysis. The same holds for the SlotAlgorithm and the sequence we constructed to calculate the size of the unoccupied area below a square.

In addition, it may be possible to apply better lower bounds on the packing than just the total area, *e.g.*, the one arising from *dual-feasible functions* by Fekete and Schepers [12].

References

1. Yossi Azar and Leah Epstein. On two dimensional packing. *Journal of Algorithms*, 25(2):290–310, 1997.
2. Brenda S. Baker, Donna J. Brown, and Howard P. Katseff. A $5/4$ algorithm for two-dimensional packing. *Journal of Algorithms*, 2(4):348–368, 1981.
3. Brenda S. Baker, Edward G. Coffman, Jr., and Ronald L. Rivest. Orthogonal packings in two dimensions. *SIAM Journal on Computing*, 9(4):846–855, 1980.
4. Brenda S. Baker and Jerald S. Schwarz. Shelf algorithms for two-dimensional packing problems. *SIAM Journal on Computing*, 12(3):508–525, 1983.
5. Ron Breukelaar, Erik D. Demaine, Susan Hohenberger, Hendrik Jan Hoogeboom, Walter A. Kosters, and David Liben-Nowell. Tetris is hard, even to approximate. *International Journal of Computational Geometry and Applications*, 14(1–2):41–68, 2004.
6. Donna J. Brown, Brenda S. Baker, and Howard P. Katseff. Lower bounds for on-line two-dimensional packing algorithms. *Acta Informatica*, 18(2):207–225, 1982.
7. Peter Brucker. *Scheduling Algorithms*. Springer-Verlag, 2nd edition, 2004.
8. Edward G. Coffman, Jr., Peter J. Downey, and Peter Winkler. Packing rectangles in a strip. *Acta Informatica*, 38(10):673–693, 2002.
9. Edward G. Coffman, Jr., Michael R. Garey, David S. Johnson, and Robert E. Tarjan. Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9(4):808–826, 1980.

10. János Csirik and Gerhard J. Woeginger. Shelf algorithms for on-line strip packing. *Information Processing Letters*, 63(4):171–175, 1997.
11. Sándor P. Fekete, Tom Kamphans, and Nils Schweer. Online square packing. In *Proceedings of the 11th Algorithms and Data Structures Symposium (WADS'09)*, volume 5664 of *Lecture Notes in Computer Science*, pages 302–314. Springer-Verlag, 2009.
12. Sándor P. Fekete and Jörg Schepers. New classes of lower bounds for bin packing problems. In *Proceedings of the 6th International Conference on Integer Programming and Combinatorial Optimization (IPCO'98)*, volume 1412, pages 257–270. Springer-Verlag, 1998.
13. Sándor P. Fekete, Jörg Schepers, and Jan van der Veen. An exact algorithm for higher-dimensional orthogonal packing. *Operations Research*, 55:569–587, 2007.
14. Gabor Galambos and J. B. G. Frenk. A simple proof of Liang's lower bound for online bin packing and the extension to the parametric case. *Discrete Applied Mathematics*, 41(2):173–178, 1993.
15. Gabor Galambos and Gerhard J. Woeginger. Online bin packing - a restricted survey. *Mathematical Methods of Operations Research*, 42(1):25–45, 1995.
16. Ronald L. Graham. Bounds on multiprocessing timing anomalies. *SIAM Journal of Applied Mathematics*, 17(2):416–429, 1969.
17. Xin Han, Kazuo Iwama, Deshi Ye, and Guochuan Zhang. Strip packing vs. bin packing. In *Proceedings of the 3rd International Conference on Algorithmic Aspects in Information and Management (AAIM'07)*, volume 4508 of *Lecture Notes in Computer Science*, pages 358–367. Springer-Verlag, 2007.
18. Claire Kenyon and Eric Rémila. Approximate strip packing. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science, (FOCS'96)*, pages 31–36. IEEE Computer Society Press, 1996.
19. Ingo Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In *Proceedings of the 2nd Annual European Symposium on Algorithms (ESA'94)*, volume 855 of *Lecture Notes in Computer Science*, pages 290–299. Springer-Verlag, 1994.
20. Steven S. Seiden. On the online bin packing problem. *Journal of the ACM*, 49(5):640–671, 2002.
21. Daniel D. Sleator. A 2.5 times optimal algorithm for packing in two dimensions. *Information Processing Letters*, 10(1):37–40, 1980.
22. Rob van Stee. Combinatorial algorithms for packing and scheduling problems, 2008. Habilitationsschrift, University of Karlsruhe.
23. A. Steinberg. A strip-packing algorithm with absolute performance bound 2. *SIAM Journal on Computing*, 26(2):401–409, 1997.
24. André van Vliet. An improved lower bound for online bin packing algorithms. *Information Processing Letters*, 43(5):277–284, 1992.
25. Deshi Ye, Xin Han, and Guochuan Zhang. A note on online strip packing. *Journal of Combinatorial Optimization*, 17(4):417–423, 2009.